A Scalable Approach for Application Layer Multicast in P2P Networks

Mourad AMAD[1] and Ahmed Meddahi[2]

[1]University of Bejaia, Algeria [2]GET/Telecom Lille 1, France

Outline

1. Introduction

- Many applications (Group communication, …) need multicast support to provide its services (IP Multicast and Application layer multicast).

- Advent of multimedia technology surge lead to excessive usage of P2P application.

➧ Combined **multicast mechanism** and **peer to peer** network is a recent challenge.

2.1. P2P overview

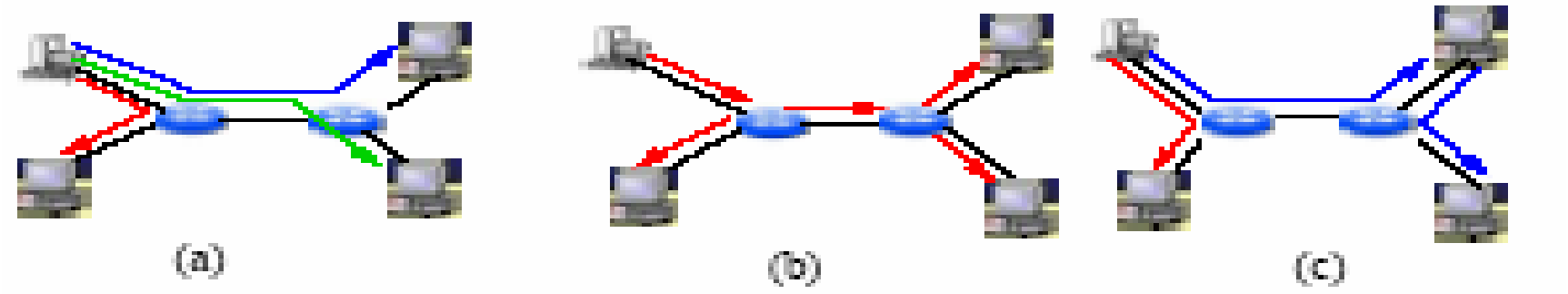
P2P systems allow mutual exchange of information and services directly between a sender and receiver.
P2P applications include :

- Sharing of large file over the Internet
- Large distributed computing
- Video on demande (VoD) applications
- Internet telephony
- P2P Media streaming applications generally based on Application Layer Multicast (ALM)

Examples of P2P networks: P4L, CAN, Chord, Pastry, Gnutella, Napster….

2.2. Application Layer Multicast

Application layer multicast refers to the implementation of multicast capability at the application layer instead of network layer.



Unicast IP Multicast Application Layer Multicast (ALM)

Examples of ALM: Narada, Nice, Scribe, TBCP, HMTP, HBM, …

3. Problems and goals

Problems	Goals
Delay, Tree depth, control overhead are always a critical optimization parameters in each ALM protocol	The optimization of the three depth from sender to all receivers gives an optimized approach
The above proposed approaches have each one its network topology	The implementation of multicast support independently of network topology gives a generic solution
The above proposed approaches does not support simultaneously multi tree construction for enabling multiparty conferencing	Enabling communication between multiple senders and multiple receivers facilitates the implementation of multiparty conferencing

The proposed approach

4.1. Primitive Approach

- Primitive Approach : Principle

- **Source Node (Ni)** : Send message Child (Ni) to all its neighboring
- **Relay Node (Nj):** At the reception of message Child (Nt)

If (Nj has accepts a similar message) then

Discard this message and sends NACK to the sender

Else

Send ACK to the sender and forwards a message Child (Nj) to all its neighboring

The primitive approach is simple but it is not necessary optimized

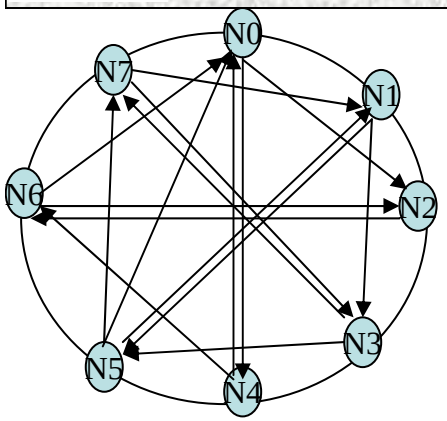
4.2. Optimized Approach: Algorithm

- Optimized approach
- **Source node (Ni):** Sends message Child (Ni) to all its neighboring

- **Relay node (Nj):** At the reception of message Child (Nt), sends message Child (Nj) to all its neighboring excepts those in set **A**, **B**, and **C where:**

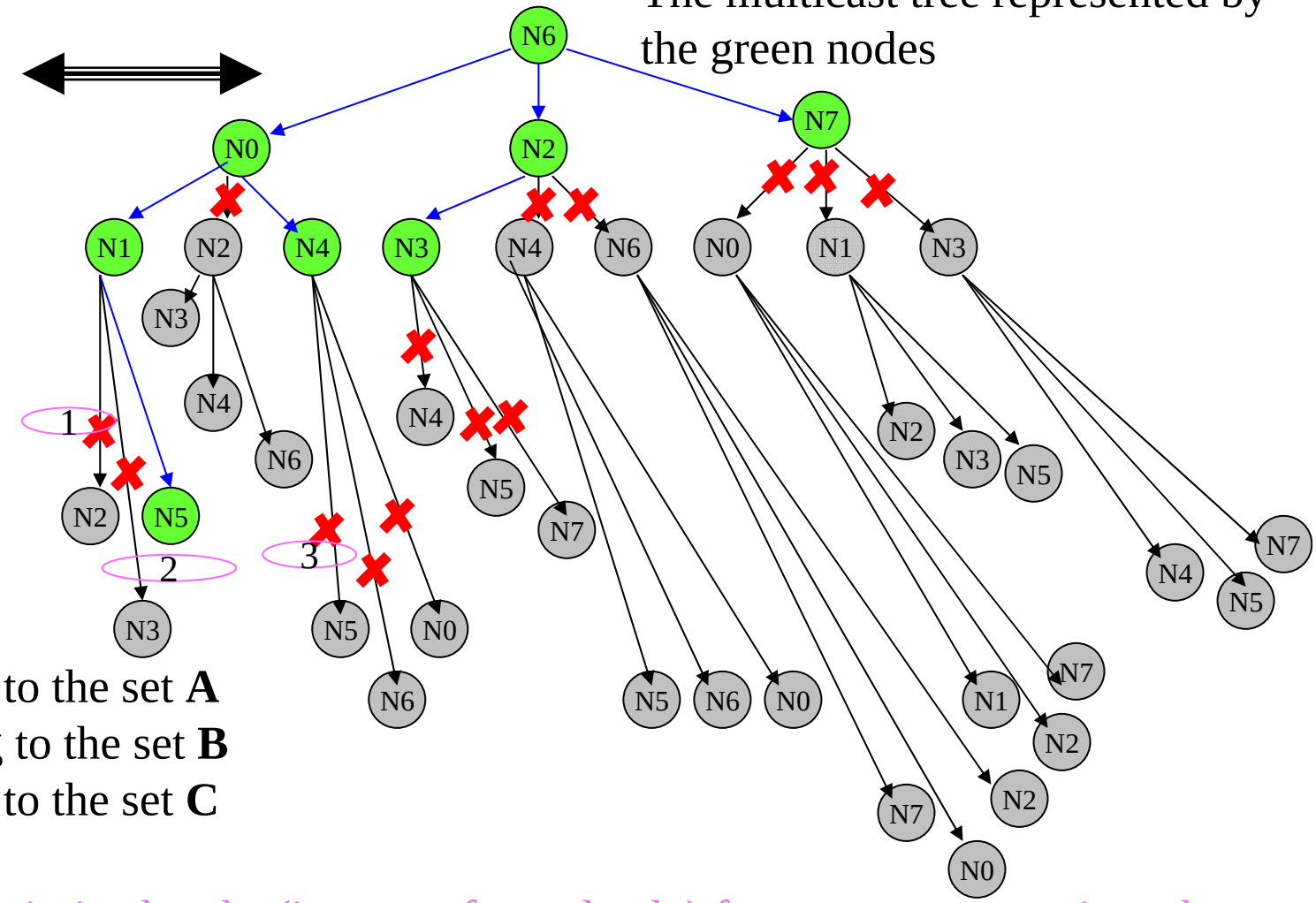
A: The set of nodes which precede Nj (**Parent**)
B: The set of nodes at the same level of Nj (**Brother**)
C: The set of nodes which is child of its brothers and those last have an identifier numerically lower than that of Nj

4.2. Optimized Approach: Example



Chord or any P2P Architecture

The multicast tree represented by the green nodes



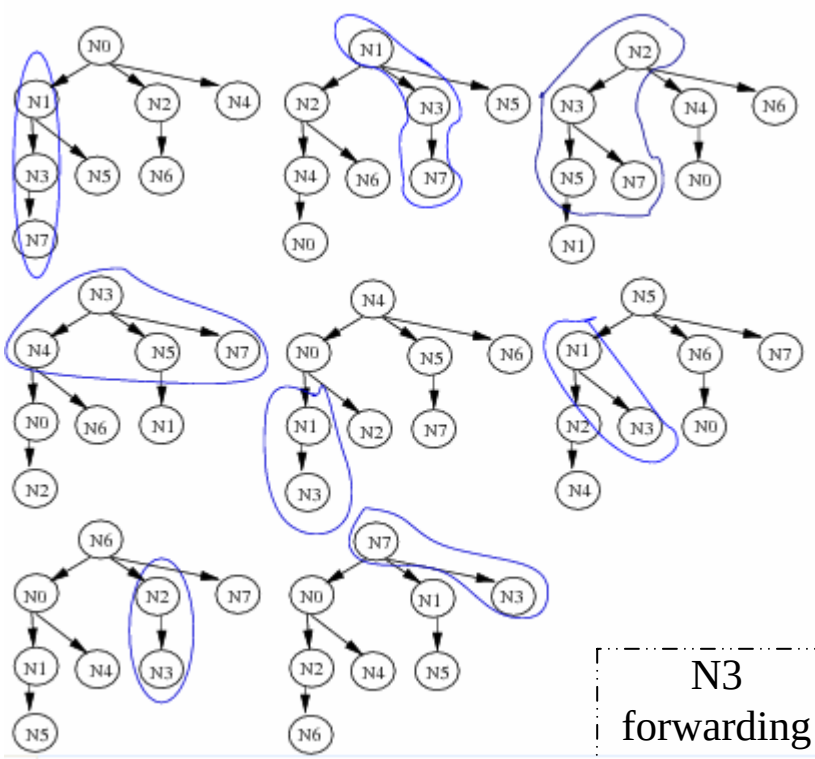
1: Node N2 belong to the set **A**
2: Node N3 belong to the set **B**
3: Node N5 belong to the set **C**

There is no other optimized paths (in term of tree depth) from source to receivers better than those one

4.3. Extension to Multi tree

- Extension to many to many application

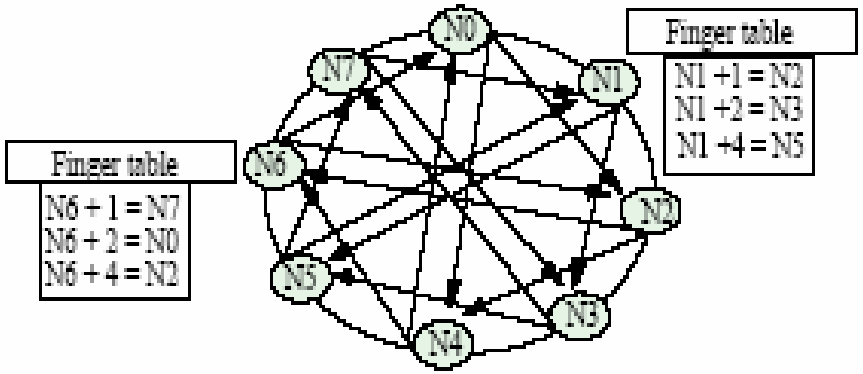
- The first entry contains the nodes from which *N3* can receives data.
  The matrix columns contain the nodes to which node *N3* forwards data



N3 forwarding matrix

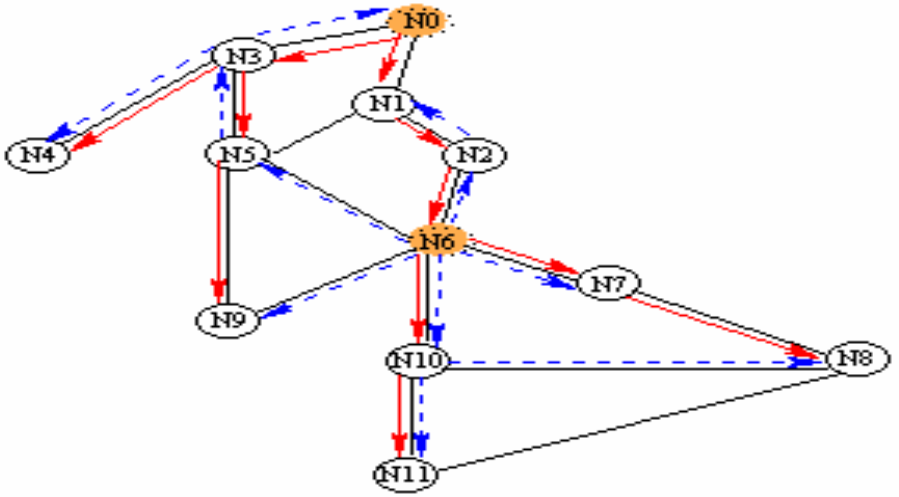
N1	N2	N7
N7	N5	Null
Null	N7	Null
Null	Null	Null

4.4. Performance evaluation



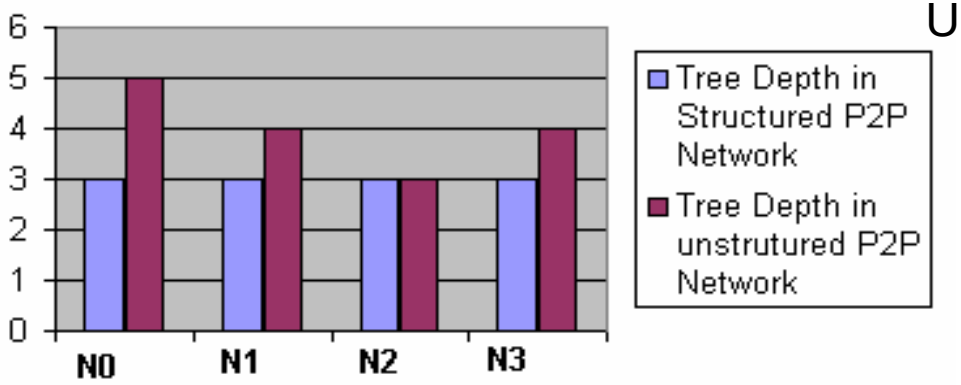
Chord architecture

Structured P2P network



Gnutella architecture

Unstructured P2P network



Tree depth in structured and unstructured P2P system for *Ni* as root

5. Conclusion & Perspective

- We have proposed a novel application layer multicast in P2P Network
- The approach is generic (*independent of network topology*)
- The approach is optimized (*The average paths from sender to receivers is optimal*)
- The approach can be easily extended into multi tree construction for multiparty conferencing applications

Future works :

- Take into consideration the **QoS** (*Quality of Service*) such as delay instead of average tree depth
- Combining a security mechanism

Thank You

Questions ?